Santa Clara High Technology Law Journal

Volume 33 | Issue 3 Article 3

2-22-2017

Open-Source Licensing and Business Models: Making Money by Giving it Away

Andrew J. Hall

OPEN-SOURCE LICENSING AND BUSINESS MODELS: MAKING MONEY BY GIVING IT AWAY

Andrew J. Hall[†]

Free and open-source software (FOSS) has become an integral part of nearly any successful business model that depends upon commercializing computing software. Whether creating consumer products that incorporate software, licensing software in exchange for customer royalty payments, or offering software-based services via the internet, providers of software products and services ("providers") will typically be required to use at least some FOSS in order to deal with third-party vendors, providers, partners, and customers. For the few providers that are not required to use FOSS, failure to take advantage of available royalty-free FOSS resources for at least some portion of the business arguably reflects a poor investment of financial and technical resources. However, while nearly all providers' businesses depend upon FOSS, some providers go further by commercializing specific FOSS projects or otherwise incorporating FOSS licensing into the delivery of their products and services.

Section I of this article addresses the fundamentals of FOSS licensing with a particular focus on characteristics of FOSS licenses relevant to commercial use of FOSS. Section II catalogs some of the common ways that companies are directly commercializing FOSS projects or incorporating FOSS licensing into their software products and services.

---

† Andrew J. Hall is a founder of Hall Law, a boutique law firm focused on software, open-source, and IP governance. Andrew regularly counsels clients on developing business models that include, depend upon, or are exclusively based in free and open-source software. Andrew also has experience counseling clients across a spectrum of circumstances on free and open-source software governance, audits, use, and compliance including hosted and distributed use of software licensed under copyleft (aka hereditary, reciprocal, and viral) licenses such as the GNU General Public License (GPL) and GNU Affero General Public License (AGPL). Prior to Hall Law, Andrew counseled clients on open-source and IP governance at Fenwick & West. Andrew also has experience in patent litigation, USPTO interference, and other patent and trademark-related disputes from Knobbe Martens. Before receiving a J.D. from the University of Chicago and after receiving a B.S. in Electrical and Computer Engineering from The Ohio State University, Andrew enjoyed a career as an applications engineer.

I. INTRODUCTION TO FREE AND OPEN-SOURCE LICENSING

Before building a business that relies upon FOSS or incorporates FOSS licenses, it is helpful to understand the different categories of FOSS licenses and how the characteristics of those categories can impact commercial use of FOSS.

A. Free, Open-Source, and Public Licensing

The terms "free software" and "open-source software" refer to software licensed under terms satisfying the specific criteria set forth by the Free Software Foundation[1] and Open Source Initiative,[2] respectively. The free and open-source software definitions are not coextensive, but nonetheless place many of the same requirements on qualifying licenses including the availability of the licensed software's source code and the recipient's rights to modify and redistribute the software. Despite the imperfect correlation, the terms "free software" and "open-source software" are often used interchangeably or referred to collectively as "open source," "FOSS," or "FLOSS."

The differences between free software and open source software are meaningful to the philosophical purpose of open-source, but neither definition meaningfully impacts a commercial use analysis. Moreover, the terms "free" and "open source" are commonly used to describe software available under a license that is neither.[3] The term "open source" in particular is often used to refer to a broader class of software that is made available (a) to the public, (b) in source code form, and (c) under the terms of a standard, royalty-free license.[4] Software satisfying this broader definition might more accurately be referred to as "public-source" software. The diagram below depicts the overlap between free, open-source, and public source software licenses and provides examples of licenses falling into each category. For the purposes of this article, "FOSS" is used to refer more generally to public source software.

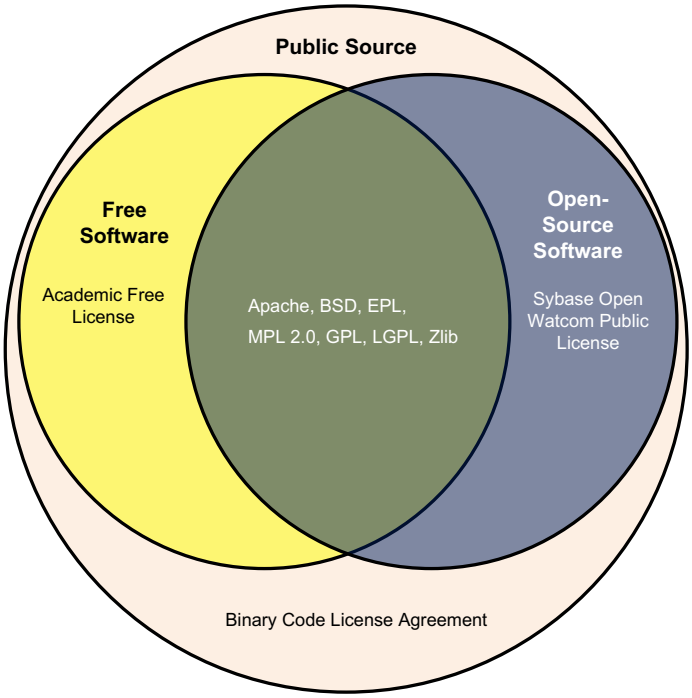
---

1. *What is free software? The Free Software Definition*, GNU.ORG (2016), http://bit.do/FreeSoftwareDefintion.

2. *The Open Source Definition*, OPEN SOURCE INITIATIVE (2007), http://bit.do/TheOpenSourceDefinition.

3. For example, software that is distributed under Oracle's Binary Code License Agreement, such as Oracle's Java EE, is often described as being "free" or "open source" despite being distributed under a license that is neither.

4. For instance, Oracle's Binary Code License Agreement, which applies to Oracle's published Java SE source code, does not qualify as a "free" or "open source" license. Many developers nonetheless refer to Oracle's Java SE software as being "open source."



The release of software in source code form under a standard, royalty-free license is one characteristic distinguishing FOSS licensing from traditional commercial licensing. A more comprehensive list of features distinguishing features is provided in the table below.

FOSS Licensing	Commercial Software Licensing
Ownership interests in the software are often distributed among many contributors.	Ownership interests in the software are typically consolidated in a single licensing entity.
Licensed to the general public under standard, non-negotiable licenses.	Licensing terms are often negotiable and can vary significantly by provider, customer, purchased products and services, and intended use.
Software is typically delivered in source form and licensed for source or binary use.	Software is often delivered and licensed for use only in binary form.

Licenses generally permit modification, subject to varying obligations and restrictions.	Licenses typically include prohibitions on reverse-engineering and modification of the software.
Licenses generally permit royalty-free redistribution of the software, subject to varying obligations and restrictions.	Licenses typically prohibit (or impose licensing fees on) the redistribution of the licensed software.
Licenses generally include explicit disclaimers of warranty and liability for downstream use of the software.	License may include intellectual property warranties or indemnification from the licensor.

B. Copyright, Patents, and Copyleft

Copyright and patents are forms of intellectual property ("IP") protection that enable commercial software distributors to place specific limitations on how their licensed software may be used by recipients. For example, IP holders may impose licensing fees on use of the software, prohibit modification or reverse-engineering of the software, or restricting the field or purpose for which the software may be used. "Copyleft" is a play on the term "copyright" and refers generally to a philosophy first espoused by the Free Software Foundation (FSF) criticizing the use of copyrights and patents to restrict the free modification, copying, and distribution of software. This copyleft philosophy is embodied in the FSF's GNU General Public License (GPL), which requires that distributors of GPL-licensed software (GPL software) make the source code for both the GPL software and any work based on the GPL software available for royalty-free use, copying, and further distribution under the terms of the GPL. FOSS licenses having a copyleft effect are also described as viral, hereditary, or reciprocal licenses. Software that must be licensed under a copyleft license due to its combination with copyleft software is often referred to as "tainted" (or, in the case of a GPL, "GPL'd"). Copyleft requirements can be at direct odds with the business models of companies that collect licensing fees for software products and services.

C. Categorizing and Describing FOSS Licenses

Despite the efforts of the FOSS community to limit the number of

FOSS licenses, there are hundreds of different FOSS licenses covering the myriad of FOSS packages available for download and use. The potential copyleft effects of different FOSS licenses tend to be of principal interest to commercial software distributors concerned about losing exclusive rights to their IP and philosophical proponents of copyleft philosophy who desire and sometimes demand compliance with copyleft requirements. A FOSS license is often categorized as either strong copyleft, weak copyleft, or permissive based on the existence and scope of the license's copyleft effect. FOSS licenses can also be designated as a prohibitive or network license based on whether specific license characteristics apply. These categories and designations are described in more detail in the remainder of this section.

D. Strong Copyleft FOSS licenses

Strong-copyleft FOSS licenses require that both the strong-copyleft software and any software that is a derivative work of (or "based on") the copyleft software be made available in source code form under the terms of the same strong-copyleft license. Nearly all copyleft FOSS licenses permit further modification and royalty-free redistribution of licensed source code software under the same copyleft terms. Many strong-copyleft licenses have a copyleft effect only on distributed derivative works of the copyleft software. Network strong-copyleft licenses (such as the AGPL) extend their copyleft effect to distributed derivative works and to at least some derivative works used in hosted environments (such as a SaaS and PaaS offerings). Commonly used strong-copyleft licenses include the GNU General Public License,[5] the Affero General Public Licensee,[6] and Creative Commons ShareAlike licenses (CC *-SA-*).[7]

Whether a derivative work has been created as a result of combining proprietary and copyleft software is ultimately a question for courts interpreting applicable copyright statutes. To date, neither the statutes nor the courts have provided clear, consistent guidance on the dividing line between derivative works and separate works that merely communicate or share information. Accordingly, an assessment of the potential impact of software combinations that include copyleft software requires familiarity with not just the applicable statutes and

5. *GNU General Public License*, GNU.ORG (2007), https://www.gnu.org/licenses/gpl-3.0.en.html.

6. *The Affero General Public License*, GNU.ORG (2007), https://www.gnu.org/licenses/agpl-3.0.en.html.

7. *E.g.*, *Attribution-ShareAlike 3.0 United States*, CREATIVE COMMONS (2017), https://creativecommons.org/licenses/by-sa/3.0/us/legalcode.

case law, but also the common opinions, policies, and practices within relevant FOSS and business communities.

E. Weak-Copyleft FOSS Licenses

Weak-copyleft FOSS licenses (also referred to as "file-level" copyleft licenses) are intended to have a narrower copyleft effect. Specifically, most weak-copyleft FOSS licenses require that distributed versions of the weak-copyleft FOSS itself, including any modifications made by the distributor, be made available to recipients in source code form under the terms of the same weak-copyleft FOSS license. Some network weak-copyleft FOSS licenses extend their copyleft effect to modified FOSS used in a hosted environment. What constitutes a modification to the FOSS is typically specified in detail within the weak-copyleft license. Many weak-copyleft FOSS licenses include explicit permission for specific combinations with software licensed under different terms (including commercial terms) without imposing a copyleft effect on the combined software. For example, most weak-copyleft licenses permit dynamic (runtime) linking of weak-copyleft FOSS libraries with proprietary software without copyleft effect on the proprietary software. Some also clearly permit static (compile-time) linking without copyleft effect. If proprietary software is combined with weak-copyleft software in a manner not authorized by the license, the weak-copyleft license can have a broad copyleft impact similar to that of a strong-copyleft license. Commonly-used weak-copyleft licenses include the GNU Library[8] and Lesser General Public Licenses (LGPL),[9] Mozilla Public License (MPL),[10] Common Public License (CPL),[11] and the Common Development and Distribution License (CDDL).[12]

F. Permissive FOSS licenses

Permissive FOSS licenses do not have a copyleft effect under any circumstance. Permissive licenses are not free from obligations, however, and, like strong-copyleft and weak-copyleft licenses, often require that recipients of the FOSS be provided with some combination

8. *GNU Library General Public License*, GNU.ORG (1991), https://www.gnu.org/licenses/old-licenses/lgpl-2.0.en.html.

9. *GNU Lesser General Public License*, GNU.ORG (2007), https://www.gnu.org/licenses/lgpl-3.0.en.html.

10. *Mozilla Public License*, MOZILLA (2017), https://www.mozilla.org/en-US/MPL.

11. *Common Public License*, OPEN SOURCE INITIATIVE (2017), https://opensource.org/licenses/cpl1.0.php.

12. *Common Development and Distribution License*, OPEN SOURCE INITIATIVE (2017), https://opensource.org/licenses/CDDL-1.0.

of attribution, copyright, and disclaimer notices, along with a copy of the applicable FOSS license. Frequently-used permissive FOSS licenses include the Apache,[13] Berkeley Software Distribution (BSD),[14] MIT,[15] and zlib[16] licenses.

G. Prohibitive Public Source Licenses

Some public-source licenses that otherwise resemble FOSS licenses place specific use restrictions on the FOSS such as prohibiting use in commercial or military applications, on unapproved platforms, or on devices providing dedicated functionality. Examples of such prohibitive licenses are the Microsoft Limited Public License (Ms-LPL),[17] Oracle Binary Code License Agreement (BCLA),[18] and Creative Commons Noncommercial Licenses (CC BY-NC).[19] These prohibitions typically disqualify the licenses from being sanctioned as either free or open-source licenses, but these standardized royalty-free licenses nonetheless share many features with FOSS licenses.

H. Network FOSS Licenses

The requirements of many FOSS licenses (including many strong-copyleft and weak-copyleft licenses) are triggered by distribution of the FOSS, which has led to the widespread belief that the use of such FOSS in hosted environments, such as SaaS offerings, does not trigger the requirements of such FOSS licenses. The ability to use GPL software in hosted solutions without being subject to the copyleft terms of the license has been acknowledged by the GPL's publisher and is often referred to as the "ASP Loophole" or "SaaS Loophole."[20] However, not all FOSS licensing requirements are triggered by distribution alone. Certain "network" licenses such as the AGPL were specifically designed to close these hosted environment loopholes, imposing FOSS obligations and restrictions not only on distributed

13. *Apache License*, APACHE (2004), https://www.apache.org/licenses/LICENSE-2.0.

14. *E.g.*, *The 3-Clause BSD License*, OPEN SOURCE INITIATIVE (2017), https://opensource.org/licenses/BSD-3-Clause.

15. *The MIT License*, OPEN SOURCE INITIATIVE (2017), https://opensource.org/licenses/MIT.

16. *zlib License*, ZLIB.NET (2017), http://www.zlib.net/zlib_license.html.

17. *Microsoft Limited Public License*, BLACK DUCK (2017), https://www.openhub.net/licenses/mslpl.

18. *Oracle Binary Code License Agreement*, ORACLE (2017), http://www.oracle.com/technetwork/java/javase/terms/license/index.html.

19. *E.g.*, *Attribution-NonCommercial 3.0 United States*, CREATIVE COMMONS (2017), https://creativecommons.org/licenses/by-nc/3.0/us/legalcode.

20. The terms "ASP" and "SaaS" are acronyms for "application server provider" and "software as a service," respectively. Both refer to software functionality provided on-demand over a network such as the internet.

FOSS but also on FOSS use provided over a network such as the internet.

II. INTRODUCTION TO FREE AND OPEN SOURCE BUSINESS MODELS

The most commonly-adopted FOSS business models can generally include one or more of the following: (a) providing products and services that complement FOSS projects; (b) offering commercially friendly licensing terms for software otherwise available on FOSS terms; (c) providing limited versions of the software under an open-source license and commercially licensing enhanced versions, plugins, or extension to the FOSS; (d) providing a FOSS platform and commercially licensing enhanced versions, plugins, extension or applications for the platform or charging a royalty for software or content distributed via the platform; and (e) providing enhanced closed-source distributions of popular FOSS projects. These business models are described in greater detail in the remainder of this section.

A. Offering Complementary Products and Services

The most commonly used open-source business model offers products, services, or combinations thereof that complement or otherwise support popular FOSS projects. The commonality of such supporting products and services can be explained, in part, by the ability of nearly anyone to offer competitive products and services for the same FOSS project. Unlike some of the business models described in the remainder of this article, providing supporting products and services does not necessarily require that the provider have exclusive rights to the delivered product.[21] The provider's lack of exclusive rights may result from the provider not generating significant software or other IP in the course of providing its products or services or because the IP that is generated is either distributed to recipients under a FOSS license or contributed back to the FOSS project complemented by the products or services.

The distinction between software products and software services, for the purposes of this article, depends upon the resources consumed by provider in delivering the product or service and, consequently, how well the business model scales. Supporting services tend to rely heavily upon human resources, such as developers and consultants. Examples of supporting services include customizing the FOSS project to meet a customer's unique needs; providing support, maintenance, development, consulting, or training services to customers that depend

21. Exclusive rights may include exclusive copyrights, patent rights, or rights associated with trade secrets.

upon a particular FOSS project; and providing auditing and legal services relating to open source. For example, Red Hat, Canonical, Novell, and others offer support, maintenance, design, and consulting services for their respective customized Linux distribution (RHEL, Ubuntu, Suse). Pivotal similarly provides consulting, development, and training services related to its FOSS software development platform, Spring. Companies like Black Duck, Palamida, and Protecode offer source code audit services that identify specific FOSS packages included in commercial software distributions. FOSS attorneys help their clients navigate risks and requirements inherent to and contractually imposed by FOSS use and adopt effective and efficient governance policies and processes for managing client use of FOSS.

While providing supporting services is the most common approach to commercializing FOSS, it also tends to be the most difficult to scale because providing additional services usually requires hiring and training additional staff. Supporting products, by contrast, tend to consume hardware and other data processing resources that scale far better. Examples of supporting products include hosting services for common combinations of FOSS (often referred to as "stacks"),[22] providing warranty and indemnification coverage for FOSS use, offering early access to updates and other software that will eventually be released under an FOSS license, and the licensing of software products that support the commercial use of FOSS. MongoLab, for example, provides hosting services for the popular MongoDB FOSS database project. MongoDB, Inc. offers commercial licenses that can include licensee indemnification for claims made against the licensee based on its use of the MongoDB database software.[23] Red Hat likewise offers its "intellectual property assurance program" to paid Red Hat subscribers.[24] Since acquiring Sourcefire, Cisco has continued the practice of granting paid subscribers early access to rule sets later released on a royalty-free basis for use with its FOSS-licensed network security software, Snort.[25] Palamida and Black Duck both offer governance software tools for scanning and managing

22. Software functionality delivered via the internet is often referred as providing software "as a service" (SaaS). Providers that purchase, host, support, maintain, and lease to customers the servers and other hardware necessary to run such software are often said to be providing hosting services. While commonly described as services, the primary resource consumed in providing *additional* services are technical (e.g., additional servers and networking equipment) rather than human. Accordingly, for the purposes of this article, such software and hosting services are considered products rather than services.

23. *See MongoDB Licensing*, MONGODB (2017), https://www.mongodb.com/community/licensing.

24. *Open Source Assurance*, RED HAT (2017), https://www.redhat.com/en/about/open-source-assurance.

25. *See Snort FAQ*, SNORT.ORG (2017), https://www.snort.org/faq.

FOSS use within a company.

B. Offering Complementary Products and Services

Another frequently adopted FOSS business model offers the same software under both FOSS and commercial terms. Unlike the FOSS products and services described in the previous section, companies adopting this "dual-licensing" or "multi-licensing" approach must generally own or have the exclusive rights to license the software offered under multiple licenses. The FOSS license offered by the software copyright holder is typically a strong-copyleft or prohibitive license unfriendly to at least some commercial closed-source uses of the software. Often, the intended use of the dual-licensed software will arguably have a copyleft effect on distributors' or hosted users' proprietary software. The copyright holder then offers a commercial license for companies who wish to use or distribute the software without being subject to the undesirable requirements of the FOSS license. Alternatively, some commercial licenses may provide access to additional products and services such as those described in the previous section. Examples of dual-licensed software with FOSS and commercial license alternatives include Oracle's MySQL (GPL-2.0), Berkeley DB (AGPL-3.0), Java EE, SE, and ME (BCLA), OpenJDK, and NetBeans IDE (CDDL-1.0, GPL-2.0 with Classpath Exception), Digium's Asterisk (GPL-2.0), and Digia's Qt (LGPL-2.1) software.

C. Open Core/Freemium

Some companies offer limited, "standard," or "lite" versions of their software under a FOSS license, while also offering enhanced or "enterprise" versions of the software under commercial terms. Commercial versions of the licensed software typically include enhanced software functionality or performance. The enhanced functionality may target specific end users of the FOSS whom the licensor hopes to convert to commercial users. Such "open-core" or "freemium" software offerings are not unlike the service models adopted by providers such as Dropbox, Pandora, LinkedIn, Evernote, and MailChimp. Examples of open-core software with enhanced commercial counterparts include Proofpoint's Sendmail, SugarCRM's Sugar, and Zimbra's Collaboration software.

D. Open Platform

Some companies provide a computing or service platform under a FOSS license. Like open core, open-platform providers often have commercial products, such as plug-ins or extensions, which

complement FOSS platforms owned or contributed to by the provider. Some platform providers also enable the delivery and use of commercial applications, services, or content subject to a surcharge collected by the platform provider. The Eclipse software development platform, for example, was originally released under a FOSS license by IBM, which still (along with many others) sells commercial plugins for Eclipse Foundation's platform. Pentaho licenses its core business analytics platform under the GPL-2.0 while offering commercial plugins and extensions authored by Pentaho and others through its Pentaho Marketplace. Google licenses portions of its Android platform under the Apache-2.0 license while commercially licensing popular Google Android application (such as Gmail) to device manufactures and collecting a percentage of fees collected for third-party application sold through its Google Play store.

E. Closed-Source FOSS Distributions

In its broadest sense, closed-source distribution of FOSS almost certainly represents the broadest use of FOSS by commercial enterprises. In fact, nearly all commercial products including software include at least some FOSS. However, for the purposes of this article, closed-source FOSS distribution refers to companies selling commercial (closed-source) versions of FOSS projects. Permissively licensed FOSS can be modified and combined with the provider's proprietary software without obligating the provider to give access to the source code for the provider's proprietary code. Companies such as Cloudera, Hortonworks, MapR, and AWS, for example, offer commercial versions of the permissively licensed Apache Hadoop software.

III. CONCLUSION

Dependence upon FOSS has become an integral part of nearly any successful business model that depends upon commercializing computing software. The few providers that are not required to use FOSS can nonetheless benefit from the available royalty-free FOSS resources. Some providers go further, integrating FOSS licensing and principles into their business models, often implementing one or more of the business models described in this article.